# Bonus Project: Bubble Sort

Sorting algorithms are some of the earliest that you typically get exposed to in Computer Science. It may not be immediately obvious how important they are, but it shouldn't be hard to think of some examples where your computer needs to sort some massive datasets during everyday operations.

One of the simpler (but more processor-intensive) ways of sorting a group of items in an array is **bubble sort**, where each element is compared to the one next to it and they are swapped if the one on the left is larger than the one on the right. This continues until the array is eventually sorted.

Check out this video from Harvard's CS50x on Bubble Sort.

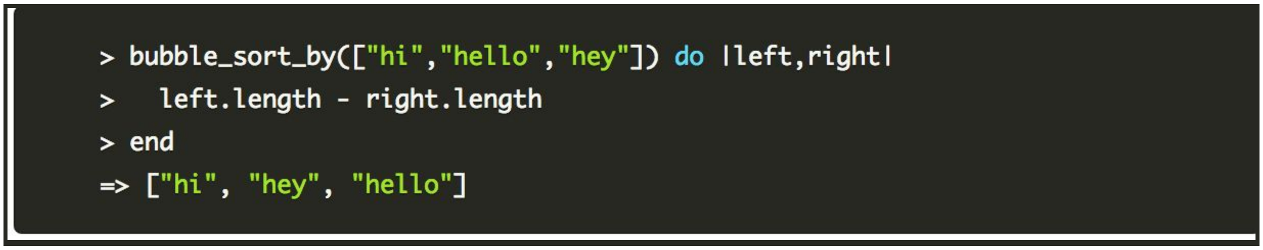
There's also an entry on Bubble Sort on Wikipedia that's worth taking a look at.

## Your Task

1. Build a method `#bubble_sort` that takes an array and returns a sorted array. It must use the bubble sort methodology (using `#sort` would be pretty pointless, wouldn't it?).

2.
   ```
   > bubble_sort([4,3,78,2,0,2])
   ```
   ```
   => [0,2,2,3,4,78]
   ```
3. Now create a similar method called `#bubble_sort_by` which sorts an array but accepts a block. The block should take two arguments which represent the two elements currently being compared. Expect that the block's return will be similar to the spaceship operator you learned about before -- if the result of the block is negative, the element on the left is "smaller" than the element on the right. `0` means they are equal. A positive

result means the left element is greater. Use this to sort your array.



```
> bubble_sort_by(["hi","hello","hey"]) do |left,right|
>   left.length - right.length
> end
=> ["hi", "hey", "hello"]
```